

Benefits and Limitations of Tapping into Stored Energy for Datacenters

Sriram Govindan, Anand Sivasubramaniam, Bhuvan Urgaonkar
Department of Computer Science and Engineering
The Pennsylvania State University
University Park, PA
{sgovinda,anand,bhuvan}@cse.psu.edu

Abstract. Datacenter power consumption has a significant impact on both its recurring electricity bill (Op-ex) and one-time construction costs (Cap-ex). Existing work optimizing these costs has relied primarily on throttling devices or workload shaping, both with performance degrading implications. In this paper, we present a novel knob of energy buffer (eBuff) available in the form of UPS batteries in datacenters for this cost optimization. Intuitively, eBuff stores energy in UPS batteries during “valleys” - periods of lower demand, which can be drained during “peaks” - periods of higher demand. UPS batteries are normally used as a fail-over mechanism to transition to captive power sources upon utility failure. Furthermore, frequent discharges can cause UPS batteries to fail prematurely. We conduct detailed analysis of battery operation to figure out feasible operating regions given such battery lifetime and datacenter availability concerns. Using insights learned from this analysis, we develop peak reduction algorithms that combine the UPS battery knob with existing throttling based techniques for minimizing datacenter power costs. Using an experimental platform, we offer insights about Op-ex savings offered by eBuff for a wide range of workload peaks/valleys, UPS provisioning, and application SLA constraints. We find that eBuff can be used to realize 15-45% peak power reduction, corresponding to 6-18% savings in Op-ex across this spectrum. eBuff can also play a role in reducing Cap-ex costs by allowing tighter overbooking of power infrastructure components and we quantify the extent of such Cap-ex savings. To our knowledge, this is the first paper to exploit stored energy - typically lying untapped in the datacenter - to address the peak power draw problem.

Categories and Subject Descriptors

C.0 [Computer Systems Organization]: General

General Terms

Design, Experimentation, Measurement

Keywords

Battery, Datacenter, Peak power, UPS

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ISCA'11, June 4–8, 2011, San Jose, California, USA.

Copyright 2011 ACM 978-1-4503-0472-6/11/06 ...\$10.00.

1. INTRODUCTION AND MOTIVATION

Power and energy consumption of datacenters have come under much scrutiny in recent times due to their contribution to cost, design and reliability constraints, and environmental concerns. In existing datacenter studies, the two terms - energy and power - are often used rather interchangeably. However, these have very different connotations - energy is the integral of power over time - and can have very different implications for datacenter design, operation, and costs. For the same energy consumption, one could have different power draw profiles, each having a different consequence on the design and reliability issues (e.g., sustained peak power correlates to the heat load) as well as on the monthly electricity bill. Some recent studies have started to look at the peak power draw problem by proposing solutions to throttle computing devices [12, 25, 40, 41, 45] and/or shift the workload peak draw temporally/spatially [17, 34]. These solutions can have adverse performance consequences depending on the workload behavior. This paper proposes a complementary solution to this problem of peak power that does not have any performance consequence, by exploiting already existing energy storage (UPS) facilities within the datacenter. It can be combined with existing throttling and workload management techniques to further reduce the peak power draw and/or the duration of this draw.

Peak power has implications on both the one-time capital cost as well as the recurring operational cost of a datacenter. The cost of designing and building a datacenter (capital expenditure or “Cap-ex”) is directly impacted by the datacenter’s peak power requirement, which determines the *provisioned* capacity of its power infrastructure (utility substations, diesel generators, uninterruptible power supply units, etc.). Current estimates suggest that Cap-ex grows on average with provisioned capacity as \$15-20/W [5, 21]. This amounts to a contribution of \$150 million to the Cap-ex of a datacenter with a provisioned peak capacity of 10MW.

The other important cost related to peak power comes in the form of the recurring (typically monthly) electricity bill paid to the utility (operational expenditure or “Op-ex”). Several datacenter studies [7, 8, 24, 38] have focused on energy reduction - reducing the KWh consumed per month - to bring down Op-ex. Utilities, however, charge datacenters in rather complicated ways, implying that energy minimization need not coincide with minimization of Op-ex. In particular, two aspects of this complexity in utility billing of datacenters are worth understanding. First, utilities often bill datacenters separately for the peak power they draw over the month (it is important to note the difference between this peak and

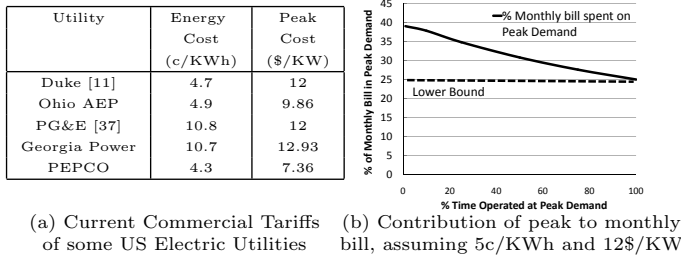


Figure 1: Importance of peak power in monthly utility bill. For (b), we assume the draw is 50% of peak, when we are not drawing the peak. As more time is spent operating near peak, the energy component grows, and the fraction of the bill due to peak drops.

the provisioned peak capacity at construction time) in addition to the energy consumed. Unlike the energy component, which is tracked over a month, this peak power component is tracked by utility companies at finer time scales (typically as the maximum of average power drawn over 15-30 minute durations). These tariffs are listed for some major US utilities in Figure 1(a). Figure 1(b) shows the contribution of the peak draw component in the electricity bill as a function of the fraction of time this peak power is drawn. For illustration purposes, we assume that the draw is 50% of the peak for the remaining time. Figure 1(b) also shows the lower bound on the percentage of the bill due to the peak component. With lesser time operating at or near peak regions, the fraction of the bill due to the peak component grows to as high as 40%. Second, the energy tariff in the utility bill itself vary with *on-peak* and *off-peak* durations (not shown in Figure 1) depending on time-of-day. The unit energy price can vary by a factor of two between these two durations [11].

The problem of reducing peak power (related to both Op-ex and Cap-ex) has analogies to several familiar resource management problems, with solution strategies sharing key commonalities. One such strategy is to limit the demand (traffic shaping in networking, growing less water-hungry crops during dry spells) to available resources, and such solutions have been adopted in the datacenter context as well - frequency scaling to throttle execution [16, 25, 40, 45] and spreading the workload over time and space [17, 34, 39]. Another strategy relies on the low likelihood that all consumers will require their resources simultaneously, allowing for statistical multiplexing gains (e.g., overbooking in airlines). Such ideas have recently been explored in datacenters for Cap-ex improvement via tighter provisioning [12, 19, 42]. However, a third common solution strategy of *buffering/storage of resources during periods of plenty/low prices ("valleys") and availing of them during periods of scarcity/high prices ("peaks")* has not been explored in datacenters. Such examples may be seen in the form of dams for water storage, packet buffers in networking, among many others. Energy storage is often an expensive proposition, making it less attractive in electrical grids. However, in a datacenter, there already exists an energy storage facility, namely the Uninterruptible Power Supply unit (UPS).

In this paper, we propose to exploit the UPS within a datacenter as an energy buffer (*eBuff*) for improving costs.

eBuff can allow for a variety of improvements in both Op-ex and Cap-ex. First, it can be used for *shaving peaks - hiding them from the utility*: *eBuff* uses stored energy in UPS batteries to reduce the peak power drawn from the utility, resulting in savings related to the peak tariff in Op-ex. Second, *eBuff* can be used to store energy when it is cheaper, which can then be used to augment the utility when energy prices go up, resulting in savings related to the variable energy tariffs in Op-ex. Third, *eBuff* can also be used for *riding peaks - hiding them from the power infrastructure*: *eBuff* can be used to overbook the power infrastructure capacity, resulting in savings related to Cap-ex. Although *eBuff* has general applicability dealing with all three of these, in this paper, we focus on the first aspect of using *eBuff* to reduce the peak power draw from utility and present our analysis and results in this context. We also discuss how *eBuff* can be adopted for reducing costs related to the other two issues. Unlike existing techniques for power cost reduction, such as device throttling, workload shifting/scheduling, etc., *eBuff* has no performance degrading consequences. Furthermore, it can (i) supplement these techniques to amplify their benefits, as well as (ii) complement them in regions where these techniques degrade performance.

Key Challenge In Using *eBuff*: UPS is an essential component of a datacenter, whose only purpose today is to become an active power source for a few seconds until the diesel generator is activated upon power outages. Most UPS units store energy (charged during power availability) using batteries, which typically have capacities to run the datacenter for several minutes. However, they are rarely used for this purpose, and usually become active for only a few seconds until the captive diesel generator kicks in. When we exploit this device for storage, it is *extremely critical to ensure that UPS battery lifetime and datacenter availability are not compromised*. Since battery usage impacts both datacenter availability (reduced residual capacity) and battery lifetime (frequent charge/discharge cycles), it is very important for a datacenter using *eBuff* to identify feasible lifetime-availability-cost trade-offs.

Research Contributions

- We conduct a detailed feasibility analysis of battery usage to arrive at operating points (*when to use? how much to use?*) constrained by battery capacity/lifetime and datacenter availability. Using a diverse set of realistic power demands, we find that *eBuff* can be very effective for cost reduction in datacenters with tall and narrow power peaks. However, *eBuff* is not a panacea - its effectiveness decreases as the peak width increases (due to battery lifetime/capacity issues), where other knobs might be preferable over it.
- Based on the insights gained, we develop a hybrid peak reduction technique that shows how *eBuff* can be cleverly augmented with the existing CPU throttling knob to realize higher peak power reduction at much lower performance impact, compared to using throttling alone.
- In a scaled-down experimental platform, we develop a series of insights regarding the cost benefits offered by *eBuff* for a range of workload peaks/valleys and SLA constraints. We find that with currently provisioned UPS capacities, *eBuff* offers 15-45% peak reduction, corresponding to 6-18% savings in Op-ex over this spectrum. We present a cost-benefit analysis of addi-

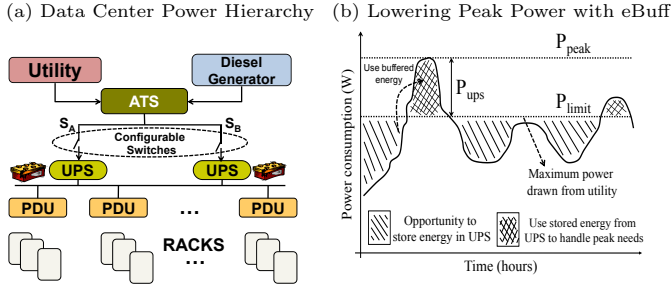


Figure 2: Incorporating eBuff in current datacenters. Proposed enhancements are shown in dotted lines in (a).

tional investments in battery capacity versus resulting savings in Op-ex, which suggests that even higher Op-ex gains are possible. Finally, we also identify potential Cap-ex gains brought by eBuff.

2. EBUFF-BASED DATACENTER: BACKGROUND AND OVERVIEW

2.1 Datacenter Power Hierarchy

Power enters the datacenter through a utility substation, which acts as its primary power source. Datacenters also employ Diesel Generators (DGs) as a secondary backup power source. An Automatic Transfer Switch (ATS) is employed to switch between these. Upon a utility failure, it takes about 10-20 seconds to transition to the DG. Uninterruptible Power Supply (UPS) systems are employed to bridge this time gap. We show these components with a 1+1-redundant UPS unit in Figure 2(a). Datacenters typically employ *double-conversion* UPS, which has zero transfer-time (unlike standby UPS) to batteries upon an outage. In a double-conversion UPS, the incoming power is first converted from AC to DC (to store in batteries) and is again converted from DC to AC (to power servers) and thus incurs a conversion loss of about 10-15%. This loss occurs *during normal operation of a datacenter and not when the energy stored in UPS batteries is used to power the datacenter*. Power from UPS units is fed to the Power Distribution Units (PDUs) which route power to racks. Racks host IT equipment (servers, switches, and storage). IT equipment these days incorporate dual power supplies for redundancy, and each of these draws power from one of the UPS units. This provides a complete redundant power path from UPS to IT equipment. Tier-3 datacenters mandate that even the power lines from the utility incorporate redundancy. Our solutions can also be adapted to N+1 redundancy, though we assume N=1 in the rest of this paper.

2.2 Role of eBuff in Optimizing Power Costs

Consider the power draw profile shown in Figure 2(b) where the peak draw is given by P_{peak} . eBuff can be used for shaving peaks for Op-ex gains. It can reduce the peak draw from utility from P_{peak} to a lower value, P_{limit} . (Recall that this peak is measured over a 15-30 minute time-scale by utilities.) As shown, whenever the datacenter power exceeds P_{limit} , it would tap into the energy stored in the UPS batteries. eBuff would store energy in the batteries when the datacenter power needs are lower than P_{limit} . It is also easy to see how eBuff can be used to incorporate variable energy prices related to Op-ex costs—by storing energy dur-

ing low-price durations and using the stored energy during high-price durations.

eBuff can also offer Cap-ex gains. Typically, the set of IT equipment that can be hosted by the power infrastructure with provisioned power capacity P_{prov} (not shown in figure) is chosen such that the likelihood of its P_{peak} exceeding P_{prov} is extremely unlikely (for both safety and performance reasons) [12, 42]. With eBuff, a datacenter can achieve similar safety/performance levels for a larger set of IT equipment connected to the same power infrastructure: this would be achieved by drawing upon the energy stored in UPS batteries to tolerate episodes of P_{peak} exceeding P_{prov} . Therefore, eBuff can allow a datacenter to more effectively overbook the capacity of power infrastructure components that are placed “above” the UPS unit in the power hierarchy. Whereas this corresponds to only the utility substations/DG in Figure 2(a), higher Cap-ex gains are likely in emerging datacenters which employ server-level UPS units (e.g., Google datacenters [18]), where one would be able to overbook the entire power hierarchy including the PDUs. To summarize, we can reduce the peak-related component of Op-ex to P_{limit} by drawing upon eBuff when P_{peak} exceeds P_{limit} (see Figure 2(b)). Similarly, we can reduce the provisioning-related peak component of Cap-ex to P_{prov} by drawing upon eBuff when P_{peak} exceeds P_{prov} . Therefore, our discussion in the rest of the paper on battery usage/lifetime constraints and datacenter availability applies to both Op-ex and Cap-ex peak optimization. In the interest of space, we only present eBuff management for realizing Op-ex savings and we ignore temporal energy price variations and Cap-ex optimization issues.

2.3 eBuff Implementation Concerns

eBuff is quite easy to implement conceptually - it only requires software-controlled switches (shown as S_A and S_B in Figure 2(a)) between the UPS and the ATS. Turning off one or both of these switches, for appropriate time durations, will register as a power outage causing the corresponding UPS to supply power. With this mechanism, we can source an average portion from the utility (upto P_{limit}) and the rest from the UPS units. We now discuss different concerns arising from employing UPS units for a purpose that is quite different from what they are intended for today.

Battery Capacity. Most UPS units come with a minimum battery capacity/runtime to sustain 8-10 minutes of operation at the peak load, even though the transition to DGs takes only a few seconds. 1+1 redundancy doubles this capacity to provide 16-20 minutes minimum duration at the peak load. In practice, the duration can be significantly longer since even during outages, the power draw is typically much lower than the peak. It is well documented that many datacenters over-provision for power by as much as 50-100% since they use face-plate ratings for provisioning, which is almost twice the average draw. As we will see in the next section (runtime chart), a lower power draw can substantially extend the battery runtime, e.g. operating at 40% of peak load triples the duration compared to operating at 100% load in Figure 3(a). Further, there are also several business reasons to provision extra battery capacity since the UPS units (which are installed soon after datacenters are built), may need to accommodate the growth of IT equipment in the datacenter over its lifetime. Finally, batteries are purchased as packs for the UPS units, leading to

a step function in capacity growth, which is not continuous. For instance, if we need higher than 8-10 minutes, we would need to go to a 20 or a 40 minute capacity in the APC family [1]. eBuff techniques can be made to exploit lower battery capacities as well, though the effectiveness would improve with higher capacities. Rather than argue for a higher capacity/provisioning, our intention here is to merely point out that eBuff is one technique to readily exploit any existing over-provisioning - whether it be in the UPS power rating with respect to the average draw, or whether it be in the battery capacity to sustain a longer duration. In our evaluations, we consider a spectrum of scenarios - from tightly provisioned systems to more conservative provisioning.

Battery runtime	Availability
1 minute	0.99999590
2 minutes	0.99999770
5 minutes	0.99999900
10 minutes	0.99999949

Table 1: Availability estimate for the datacenter shown in Figure 2(a).

the impact of UPS battery usage on datacenter availability, we build a continuous-time Markov model. This model uses available failure/recovery rates of utility, DG and, UPS units, captures battery runtime as discrete charge states, and incorporates UPS battery charge/discharge events and DG transition time. In the interest of space, we do not present our model and we will refer to our technical report [20] for details. We present the availability of the datacenter in Figure 2(a) for varying battery runtimes in Table 1. We find that even a battery runtime of just one minute achieves five 9's of availability. With higher runtimes, however, the availability improvement slows down. Specifically, five minute runtime realizes the same availability (upto six 9's) as ten minutes. We employ eBuff such that it always leaves five minute reserve charge in batteries to allow safe DG transitions. For alternate availability stipulations, a datacenter may choose a corresponding reserve charge level for eBuff.

UPS Battery Lifetime/Lifespan. A final concern is the possible lowering of UPS battery lifetime due to more frequent charge/discharges. Note that this concern applies only to the battery packs within the UPS. The rest of the UPS circuitry is unaffected by eBuff mechanisms and is typically replaced once every 10-12 years independently of the state of the batteries. Battery lifetime issue is extensively discussed in the next section, and we show how we can impose a lower bound on lifetime and constrain its use accordingly. Though the framework is general, we use a lifetime lower bound of 4 years in our experimental results, which is in line with the depreciation of IT equipment in many organizations. It is to be noted that regardless of eBuff, lead-acid batteries inherently have a lifespan of 3-5 years for numerous reasons such as grid corrosion, dry-out, etc. [32].

3. UPS CHARACTERISTICS AND IMPLICATIONS

We examine two important aspects - runtime and lifetime behavior - of a UPS unit. Although we use numbers specific to lead-acid batteries that are most prevalent in datacenters today, all these characteristics apply with slight quantitative

Datacenter Availability. Since UPS is crucial to the continued operation of datacenter, it is important to ensure that UPS discharges with eBuff does not degrade datacenter availability. To understand

the impact of UPS battery

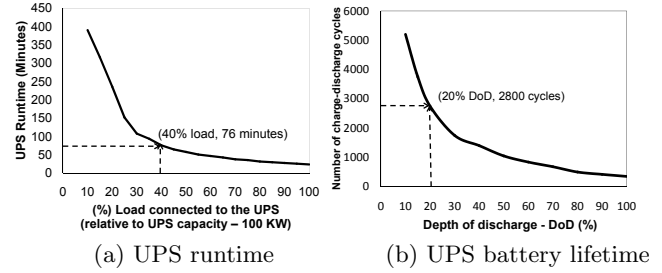


Figure 3: Runtime chart shows the amount of time the UPS can supply power at a specified load level. Lifetime chart shows the number of charge/discharge cycles sustainable for different DoD levels.

variations to other battery technologies. In the following discussion, we use examples whenever necessary with data from a 100 KW APC UPS, though the methodology is general and can be applied to even larger units (which simply use more battery cells). This UPS allows battery packs that can sustain 9, 24, 41, ..., minutes of operations at the highest load level (100 KW). Examples below use a 24 minute battery pack, allowing up to 48 minutes with 1+1 redundancy.

Discharge Behavior of a UPS. The UPS "runtime" for a load level (or power drawn from UPS) measures the time over which the UPS discharges starting from a fully charged state, until it does not have sufficient charge to sustain the required power draw. In Figure 3(a), we present the runtime chart of the 100KW UPS (chart available on APC Web site). We see that the runtime does not scale linearly with the load due to high energy losses at higher load levels. For instance, the UPS shown in Figure 3(a) runs for 236 minutes at 20% load but has a runtime of only 76 minutes (instead of 118 minutes) when the load is doubled to 40%.

UPS Lifetime. This depends on several factors including: (i) number of times it undergoes charging/discharging, (ii) extent of discharge during its operation, and (iii) issues such as grid corrosion, dry-out, etc. [32]. Each charge/discharge cycle causes active material (e.g. lead) to be shed from the electrode plates of the batteries, gradually reducing its amount and hence the lifetime [28]. For a given cycle, the "depth of discharge" (DoD) of a battery represents the fraction of total battery capacity discharged from the battery when recharging begins. DoD is expressed as a percentage of the maximum battery capacity. For example, a fully discharged battery would be described by a DoD of 100%. UPS lifetime is expressed as the number of cycles ($N(x)$) it would last on average before becoming inoperable (thus necessitating replacement of batteries) for a given value of DoD (x). Lifetime charts are typically provided by battery manufacturers based on rigorous lab measurements [28, 31]. This is pictorially shown in Figure 3(b). For illustration, the figure shows one point indicating that if this UPS is always recharged upon reaching an average DOD of 20%, it is expected to last 2800 such cycles, i.e. lifetime of 4 years assuming 2 discharges per day to a DoD of 20%.

3.1 Implications on eBuff Operation

We express the above implications as constraints that must hold each time an eBuff discharge duration begins. We define a discharge duration Δ_{ij} as a contiguous period

of time $[t_i, t_j]$ during which a constant amount $P_{ups}(\Delta_{ij})$ of power is drawn from the UPS, with the remaining $P_{limit} = P_{peak}(\Delta_{ij}) - P_{ups}(\Delta_{ij})$ coming from the utility (as explained in Section 2.2). Let us denote by $D_{start}(\Delta_{ij})$ and $D_{end}(\Delta_{ij})$ the DoD at the start and end of this discharge duration, respectively. Let $R(P_{ups}(\Delta_{ij}))$ denote the maximum duration for which $P_{ups}(\Delta_{ij})$ units of power can be drawn from the UPS (obtained from Figure 3(a)). We have, $D_{end}(\Delta_{ij}) = D_{start}(\Delta_{ij}) + (\frac{(t_j - t_i)}{R(P_{ups}(\Delta_{ij}))} \cdot 100)$. We consider eBuff operation over a finite time horizon H , say a day.

Capacity Constraint. eBuff must always have enough battery capacity at t_i , given by $D_{end}(\Delta_{ij}) < 100\%$.

Battery Lifetime Constraint. We derive a lifetime constraint to ensure that eBuff satisfies specified battery lifetime $Life_{ups}$. Recall that $N(x)$ denotes the average number of charge/discharge cycles for a battery before it becomes inoperable when discharged to a DoD level of x (see Figure 3(b)). Assuming a uniform rate of degradation in lifetime, the number of allowable discharges to a DoD of x is at most $\frac{N(x) \cdot H}{Life_{ups}}$ during H . The lifetime charts of UPS units only capture a limited amount of information about the impact of discharges on battery lifetime. First, they only capture the impact of a discharge duration for a fully charged UPS, i.e., for $D_{start}(\Delta_{ij}) = 100$. Second, lifetime charts do not provide a way to combine the impact of two or more discharge durations that leave the batteries at different DoD levels $D_{end}(\cdot)$. Consequently, we make the following conservative estimates: (i) even for $D_{start}(\Delta_{ij}) < 100$, we estimate the impact on lifetime based on assuming $D_{start}(\Delta_{ij}) = 100$, and (ii) if D_{max} is the maximum DoD during H , we assume $D_{end}(\Delta_{ij}) = D_{max}$ for all Δ_{ij} in H . This leaves us with the following upper bound on the number of discharge durations $n(H)$ allowed during H : $n(H) \leq N(D_{max})(\frac{H}{Life_{ups}})$.

Availability Constraint. Recall from Section 2.3 that based on our Markov model for datacenter availability, we always leave at least 5 minutes of reserve charge in UPS batteries. Let us denote by D_{avail} the DoD corresponding to this duration. That is, for a UPS battery with a runtime of 15 minutes, D_{avail} would be set to 66%. Therefore, we need for all discharge durations Δ_{ij} : $D_{end}(\Delta_{ij}) \leq D_{avail}$.

3.2 Understanding Our eBuff Constraints

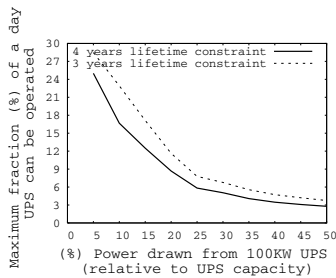


Figure 4: The maximum amount of time UPS can be operated per day at specified power draw for lifetime constraints of 3 and 4 years.

For an intuitive understanding of the above constraints, consider eBuff operation over the duration of a day ($H = 24$ hours). For different power draw (P_{ups}) from the UPS, we vary the number of discharge cycles per day and calculate the maximum amount of time eBuff can be operated per

day within lifetime constraints of 3 and 4 years. We present results for a 1+1 redundancy configuration of the 100 KW APC UPS in Figure 4 for which a 5 minute reserved headroom (to ensure availability as in Section 2.3) corresponds to $D_{avail} = 80\%$. This curve illustrates the trade-offs between how much we want to discharge the batteries versus how long we want to use the batteries per day. E.g., for $P_{ups}=20\%$ an excess (over and beyond P_{limit}) power draw of 20 KW can be sourced from UPS for at most 125 minutes (around 2 hours) a day with a lifetime constraint of 4 years.

4. PEAK SHAVING ALGORITHMS FOR OP-EX REDUCTION

Reducing peak demand has a linear impact on the utility bill. However, the knob used to reduce this demand can have other consequences. For instance, when using battery, there can be energy wastage due to charge/discharge cycles. CPU power state modulation, another well-known knob, can also reduce peak power, but can degrade performance. In the rest of this paper, we refer to these knobs as “battery” and “throttling”. We consider the problem of minimizing workload peak power consumption using the above two knobs with the following three constraints, (i) UPS constraints on capacity, lifetime, and availability (as developed in Section 3.1), (ii) power/performance trade-offs offered by CPU throttling states in our hardware for the given workload (as in Table 2 for the SPECjbb workload, which is discussed in the next section), and (iii) constraints describing workload SLA. This is a min-max problem similar to those studied in other contexts [26]. Whereas workload prediction is an essential component of a solution for such min-max problems, it is orthogonal to our research focus in this paper. Therefore, we focus on the design of an offline algorithm that assumes knowledge of the workload’s power demands (for a time horizon $H=1$ day). A large body of prior work exists on characterizing and predicting realistic datacenter workloads [9, 27], and exploring the efficacy of such prediction techniques on eBuff is part of our future work.

Our generic algorithm is provided as input a time-series $\{P_{demand}(t)\}_{t=1}^{t=H/b}$, which we refer to as a “Demand Curve”. Its elements represent datacenter power demands averaged over non-intersecting windows of length b each, for the time horizon H . The algorithm outputs the feasibility of achieving the given P_{limit} as well as when/how battery and CPU throttling should be used for achieving it. *The main idea behind the algorithm is to iteratively go from the tallest to the smallest peak(s) in $P_{demand}(t)$ to see whether a knob can be used to shave off that(those) peak(s) to the height of the next higher peak(s).* This stems from the observation that the highest peak determines the billing, and shaving one peak to a height below that of another (which is not shavable) does not help save costs.

If the knob that is to be used is the battery, then we need to ensure that (i) there is enough capacity in the battery for the appropriate level of shaving, (ii) the number of times that it is discharged over H is less than or equal to $n(H)$ to ensure minimum lifetime guarantees, and (iii) no discharge takes the battery below D_{avail} to ensure availability stipulations. The iteration stops when any of these constraints (refer section 3) cannot be met. Though not discussed here, our algorithm also accounts for the power spike induced due to UPS charging process, immediately after connecting the UPS back to utility followed by a discharge.

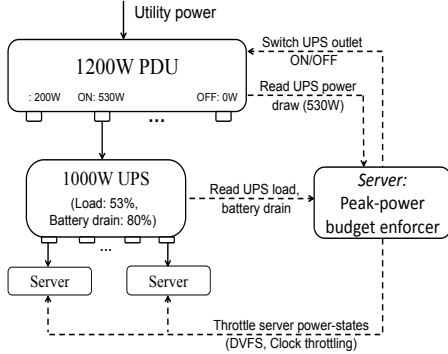


Figure 5: Experimental setup using 3 Servers, 1000W APC UPS, and a Raritan 1200W PDU.

If the knob is CPU power state modulation, the algorithm looks up the performance-power model (as in Table 2 discussed in the next section for our workload) to find the best performance state to achieve the required peak shaving. The iteration stops if the resulting performance does not meet SLA stipulations, or if there is no power state to provide the required power reduction.

5. OUR EXPERIMENTAL PLATFORM

We use a scaled-down experimental setup (Figure 5) consisting of three DELL servers with two Intel Xeon 3.4GHz processors each, running RedHat Linux 5.0, to illustrate our approach. The face-plate rating of these servers is 450W. The idle power consumption of our servers is around 160W and the peak power that we attain on a server with our workload is 320W. The dynamic power consumption can be modulated with several active power states including 4 DVFS states (P-states: 3.4GHz, 3.2GHz, 3.0GHz, and 2.8GHz, each at a different voltage level) and 8 clock throttling possibilities (T-states: 12.5%, 25%, ..., 100%). These servers are directly connected to a 1000W UPS from APC [1] which, in turn, is connected to one of the outlets of a 1200W PDU. The UPS is capable of reporting its load and remaining battery runtime over an RS232 serial interface. Our PDU is capable of dynamically switching ON/OFF the power supply to the UPS (switches shown in Figure 2(a)) in response to SNMP commands it receives over Ethernet. We employ this functionality of the PDU to control when our UPS discharges. Our PDU can report power draw from the wall socket once every second, thus capturing the utility power load for our setup. We use a separate machine (“Peak Demand Enforcer”) to send throttling commands to each server when necessary and to remotely turn ON/OFF UPS. It is important to note the following two differences between our experimental setup and a real datacenter environment:

(i) While the capacity of our UPS limits the number of servers that can be connected (3 in this case), unlike a real datacenter where hundreds/thousands of servers are powered by the UPS, we can still study peak power reduction relative to this scaled down version allowing us to consider two representative points of datacenter operation - one where the peak draw is around 64% of the provisioned capacity (2 server experiment called S_1) and another where we get close (96%) to the provisioned capacity (3 server experiment called S_2). Note that in the latter case, we are going well-

Power state (GHz, Clk.)	Performance Degradation (%)					Power Reduction (%)				
	20	40	60	80	100	20	40	60	80	100
3.2, 100%	0	0	1	4	8	0	2	5	6	6
3.0, 100%	0	0	2	6	13	0	5	10	12	15
2.8, 100%	0	1	4	9	18	2	8	15	18	22
2.8, 87.5%	2	4	8	13	20	3	9	16	19	24
2.8, 75%	6	8	13	17	23	3	11	17	21	27
2.8, 62.5%	9	13	19	24	26	4	13	19	24	31
2.8, 50%	12	19	23	27	32	4	14	23	28	33
2.8, 37.5%	19	30	38	44	50	4	16	27	30	36
2.8, 25%	26	43	55	65	70	5	18	29	33	41
2.8, 12.5%	36	60	72	84	90	6	20	32	38	45

Table 2: Power consumption and performance degradation of SPECjbb at workload intensities of 20%, 40%, ..., 100% for different CPU power states. These numbers are relative to the highest power state (3.4GHz, 100% clk.) where each tuple denotes (DVFS state, Clock Throttling state). First 3 rows are DVFS (P-states) and the latter 7 rows are Clock throttling (T-states) at the lowest DVFS state.

beyond the face-plate rating (450W), and have aggressively “over-booked” the UPS power capacity.

(ii) Since we are mimicking the behavior of a 1+1 redundant UPS configuration with a single unit in our setup, we consider batteries with 40 (20+20) minutes (called U_1) and 20 (10+10) minutes (called U_2) of runtime at the highest load, and present results for both. Without a large-scale study of several UPS units spanning several years, one cannot run meaningful lifetime or availability measurements, and we have to simply rely on analytical constraints explained in the previous section. We set D_{avail} as 75% for U_1 and 50% for U_2 (corresponding to 5 minutes of reserve capacity) and $Life_{ups}$ as 4 years for all our experiments.

Workloads. We use SPECjbb [43], a 3-tiered benchmark which emulates server-side Java applications. This application can be configured to run at various workload intensity levels expressed in percentages. The performance metric of SPECjbb is its throughput expressed as the number of transactions handled per second. We run SPECjbb on one of our experimental servers and report its power consumption and throughput at various workload intensities normalized to the highest power state in Table 2. Each server exhibits a dynamic power range between 160-320W, and we replicate the same workload on all the servers in our experiments. The load on the UPS can thus range between 480-960 W with 3 servers and between 320-640W with 2 servers. The table shows that we can reduce peak power demand by as much as 45% with power state modulation, but this comes with a severe performance penalty of 90%.

Using SPECjbb, we construct four representative power demand curves ($P_{demand}(t)$) that are shown in Figure 6(a). Our demand curves are designed to represent the diverse peak characteristics that we observe with four type of real-world workloads: (i) ‘Flash’ has tall and narrow (15 minutes) peaks that are caused by flash crowds seen by many video-streaming/e-commerce applications [30], (ii) ‘TCS’ has moderately tall peaks lasting a little longer (1 hour), corresponding to a production datacenter workload of TCS, an IT services company [44], (iii) ‘Google’ has shorter but wider (3 hours) peaks similar to the Google cluster trace [23], and, (iv) ‘MSN’ has tall and very wide (8 hours) peaks similar to the Microsoft messenger workload [9]. We capture these

different power characteristics in our demand curves by varying the SPECjbb workload intensity every $b=15$ -minutes at all servers over a $H=1$ day window. For example, a power draw of 850W in the demand curve corresponds to running SPECjbb on 3 servers at the 80% intensity during that 15-minute window. We set the peak width and height of our demand curves based on the characteristics of the representative workload. *Flash*, *TCS*, *Google*, and *MSN* have peak heights that follow a Pareto distribution whose mean is 95%, 85%, 75% and 95% of the maximum power demand and their peak widths are 15 minutes, 1 hour, 3 hours, and 8 hours, respectively. We assume the number of peaks to be 5, 3, 2, and 1, respectively, for the four workloads, spaced evenly during the 1-day horizon. The power draw is assumed to be 50% of the maximum power demand for the non-peak durations of the demand curves.

UPS config.	1000W UPS [1], $Life_{ups}=4$ yrs
UPS peak power headroom	S_1 (2 Servers = $(1000-320*2)/1000 = 36\%$), S_2 (3 Servers = $(1000-320*3)/1000 = 4\%$).
Battery config.	U_1 (20+20 minutes), U_2 (10+10 minutes). $D_{avail}=75\%$ for U_1 and 50% for U_2 .
Workload (SPECjbb [43])	<i>Flash</i> (Tall and Narrow peaks), <i>TCS</i> (Moderate peak width/height), <i>Google</i> (Shorter and Wide peaks). <i>MSN</i> (Tall and Wide peaks).

Table 3: Configuration parameters

Figure 6 (a) shows the aggregate demand curve across all servers for the 3 server configuration. For clarity, we do not present *MSN* in the figure, which just has a single peak per day of about 900-950W lasting for 8 hours. A 2/3 scaling of these demand curves gives the aggregate for the 2 server configuration. Finally, we use a wide range of SLA specifications for evaluating the performance impact of the 'throttling' knob, with our metric being tolerance factor in the 95th percentile SPECjbb throughput.

Table 3 summarizes our design space.

6. EXPERIMENTAL EVALUATION

We have presented two broad knobs - UPS battery and Throttling - to be employed by our peak reduction algorithm. We refer to these as **Only Battery** and **Only Throttle**, and use these to evaluate the effectiveness of these two knobs (in isolation) for reducing peak power. We present results for *Flash* and *Google* with P_{limit} of 650W and 600W, respectively, using *Only Battery* and *Only Throttle* in Figures 6(b) and (c). We use configurations S_2 and U_1 for this result. The *modified* demand curves in the figure indicate the power profile after applying the knobs. For clarity, we zoom in on a smaller time duration in these figures. With throttling as the only knob, *Flash* requires that the power state (2.8Ghz, 50% clk.) be used during the peak to achieve the desired P_{limit} of 650W. *Flash* has narrow and tall peaks (CPU utilization close to 90%) and, therefore, suffers a performance degradation of about 33%. With battery as the only knob, however, there is no impact on performance, while adhering to the 4-year lifetime constraint - undergoing 5 discharges per day, each with DoD of less than 10%. On the other hand, *Google* (wider and shorter peaks) suffers little performance degradation with *Only Throttle* - a mere 5% at (2.8Ghz, 87.5 clk.) - since its peaks do not saturate the server CPUs. However, battery as the only knob does not achieve the desired P_{limit} of 600W (see Figure 6(c)) within the lifetime constraint, and is able to only get to 685 W. Wider peaks

lead to higher DoD levels, causing lower lifetimes, and hence *Only Battery* is not as effective in such scenarios if we set lifetime bounds. These examples illustrate that the efficacy of a knob depends on workload characteristics.

Given the relative merits of each knob across these two diverse demand curves, we develop a simple hybrid approach, and refer it as *Hybrid* which combines both these knobs. *Hybrid* uses the battery knob up to the extent allowed by the lifetime/availability constraints and then uses the throttling knob, while adhering to the SLA constraints. *Hybrid* adjusts the fraction of peak shaved using these two knobs depending on the SLA and battery constraints. We now evaluate the efficacy of *Hybrid* on reducing peak demand.

Improved performance using eBuff. Figure 7 shows the performance degradation (in SPECjbb throughput) for *Flash*, *TCS*, *Google*, and, *MSN* when subjected to different degrees of peak reduction using *Only Throttle* and *Hybrid*. We use U_1 and S_2 configuration for these results. It can be seen that, as we move to increasing levels of peak reduction, *Only Throttle* suffers from very high performance degradation for all our workloads. This is because, for achieving higher levels of peak reduction, *Only Throttle* have to use the clock-throttling states, which significantly degrades performance (see Table 2). *Hybrid* helps to either completely eliminate the use of clock-throttling states (in case of *Flash* and *TCS*) or reduces the extent of its usage (for *Google*), resulting in improved performance. For example, for *Flash*, *TCS*, and *Google*, whereas *Only Throttle* suffers performance degradation between 50-80% for realizing 40% peak reduction, *Hybrid* incurs less than 10% performance degradation. We see in Figure 7(c) that neither *Only Throttle* nor *Hybrid* is able to achieve 40% peak reduction for *Google*. This is because, the SPECjbb workload becomes unstable and ceases to run, when subjected to *Only Throttle* (87.5% clock throttling for 40% peak reduction). Also, *Hybrid* offers only up to 35% peak reduction within the battery operation constraints.

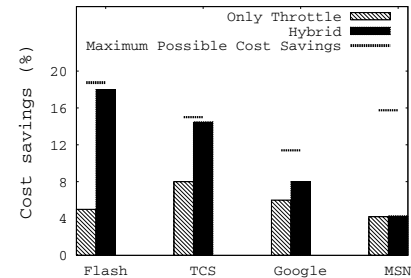


Figure 8: Savings in bill due to peak reduction offered by *Only Throttle* and *Hybrid* for 10% SLA.

It can also be seen that *MSN* with 8 hours of peak (due to very high depth of discharge and limited battery capacity) does not get much help from the battery knob and therefore *Hybrid* is similar to *Only Throttle*. Though not shown in the figure, *Only Battery* is able to reduce, 35%, 20%, 10% and 1.5% of peak for *Flash*, *TCS*, *Google*, and, *MSN* respectively, within its lifetime/capacity constraints. In general, *Hybrid* is able to realize high levels of peak reduction at much lower performance impact than *Only throttle*.

Cost Savings. We assume energy priced at 5c/kWh with the peak tariff given as \$12/kW, comparable to tariffs of

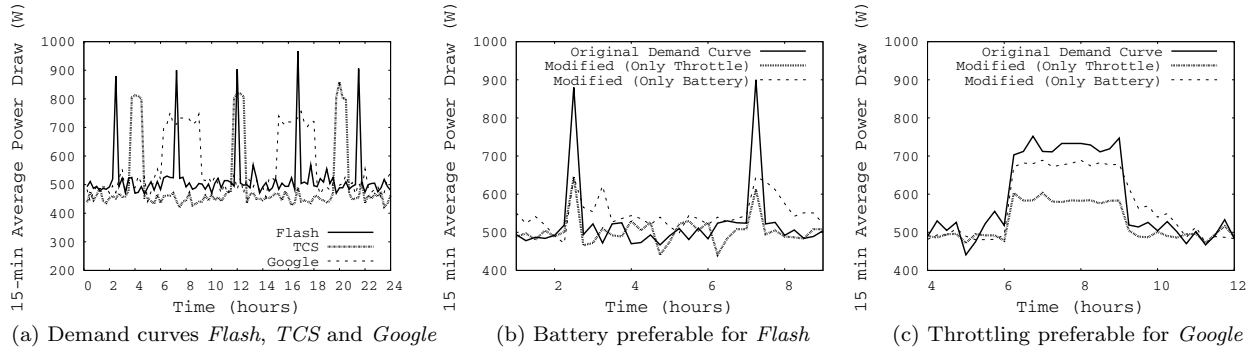


Figure 6: (a) Demand Curves. For clarity, *MSN* is not shown—it has a single peak per day of about 900-950W lasting for 8 hours. (b) *Flash* with $P_{limit}=650W$. 5 DoDs of less than 10% each with *Only Battery*. *Only Throttle* requires CPU state (2.8 GHz, 50% clk.) during peak and suffers a performance degradation of 33% (c) *Google* with $P_{limit}=600W$. 2 DoDs of 20% each, and unable to attain P_{limit} with *Only Battery*. *Only Throttle* requires CPU state (2.8 GHz, 87.5% clk.) during peak and suffers a performance degradation of only 5%.

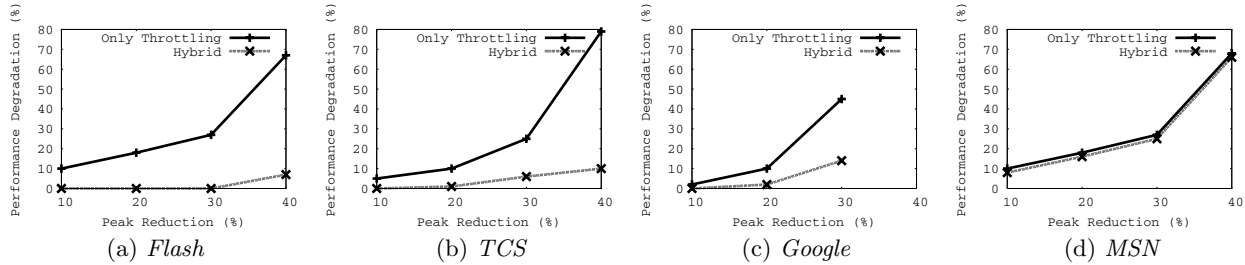


Figure 7: Performance degradation as a function of peak reduction for the workloads we consider. *Hybrid* that combines battery with throttling offers much better performance compared to *Only Throttle*. *MSN* with tall and wide peak does not get benefited from the battery knob due to limited battery capacity and lifetime issues.

Duke/Ohio AEP. We empirically measure energy wastage due to UPS charge/discharge process and find it to be 0.20-0.28 times the overall energy discharged from our UPS. Accounting for this wastage, Figure 6 shows the cost savings offered by *Only Throttle* and *Hybrid* for the 10% SLA scenario. We compare these with the maximum attainable cost reduction for each demand curve (calculated by hypothetically shaving the entire peak with an infinite battery with no energy wastage). We find the contribution of energy wastage to be less than 1% of the savings due to peak reduction, thus has a negligible impact. Consequently, the cost savings graph has a direct correlation with the peak reductions. *Hybrid* comes quite close to the ideal cost reductions possible (18%, 15% and 8% for *Flash*, *TCS*, and *Google*, respectively) as shown in the figure, except for *MSN* whose peak is too wide and tall for the battery to have any effect. In fact, for *MSN*, throttling couldn't offer much help either (only 4% savings), due to its high CPU utilization and the assumed SLA constraint of 10%. Note that the cost savings due to peak reduction crucially depend on the fraction of server idle power to the actual peak draw (since idle power affects average energy). If we assume a server where idle power contributes to only 25% of peak (unlike 50% on our experimental servers), the cost savings would increase to 27%, 26%, 15% and 6% for *Flash*, *TCS*, *Google*, and, *MSN* respectively, for the 10% SLA constraint.

It is harder to quantify Cap-ex savings due to lack of information about the power infrastructure component costs. An alternate way to characterize Cap-ex gains is to consider the following “dual”: how many additional IT equipment can be safely connected to the same power infrastructure [12, 19]

(such that the episodes of P_{peak} exceeding P_{prov} are still negligible). The peak reductions offered by *Hybrid* for 10% SLA (45%, 40%, and, 25% for *Flash*, *TCS*, and, *Google* in Figure 7), can translate in to attaching one more server to our PDU (assuming UPS is placed below the PDU in the power hierarchy, say at server level). For example, we are now able to connect 4 servers (as opposed to only 3) to our PDU. Although the overall power consumption of the 4 servers may exceed the capacity of the 1200W PDU, *Hybrid* shave peaks above $P_{prov}=1200W$, allowing operation with this overbooked PDU.

Impact of Provisioning Practices. We consider two aspects of UPS provisioning existing in current datacenters: (i) the potential peak power draw offered by the UPS relative to the actual peak draw (S_1 vs. S_2 where the latter draws 50% higher power at the peak compared to the former and leaves only 4% headroom in provisioning), and (ii) the runtime offered by the UPS batteries at the peak load (U_1 vs. U_2 where the former can run twice as long as the latter). We use CP_i to denote a configuration pair of UPS and server configurations. $CP_1:(U_1, S_1)$ denotes the most conservative provisioning (best for peak reduction) and $CP_4:(U_2, S_2)$ denotes the most aggressive provisioning (worst for peak reduction). The other two configurations, $CP_2:(U_1, S_2)$ and $CP_3:(U_2, S_1)$ lie in between, where CP_2 has a higher bias favoring the battery (since it has twice the battery capacity compared to the other whose peak load has dropped only by a third). We study the impact of *Hybrid* and compare its peak reductions with those obtainable for *Only Throttle* (with a 10% SLA constraint) for *Flash* and *Google* in

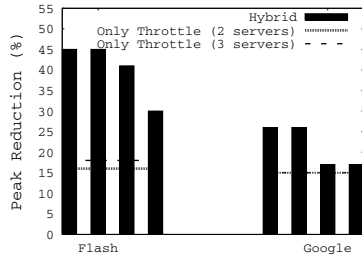


Figure 9: Impact of provisioning on peak power reduction using *Hybrid* (compared to *Only Throttle*) with 10% SLA. Each group of 4 bars corresponds to CP_1 , CP_2 , CP_3 , and CP_4 from left to right.

Figure 6. Note that the horizontal lines in these graphs correspond to peak reduction for *Only Throttle* with 2 (S_1) and 3 (S_2) servers, since *Only Throttle* results will not be any different between U_1 and U_2 . As expected, when we proceed from most conservative (CP_1) to most aggressive provisioning (CP_4), peak reduction offered by *Hybrid* decreases. However, even with aggressive provisioning practices (CP_3 , CP_4), *Hybrid* achieves up to 30% reduction in peak for *Flash* and *TCS* (*TCS* not shown in figure). For *Google*, which is less favorable to the battery, *Hybrid* still achieves 17-25% reduction in peak, higher than what is achieved with *Only Throttle*. *MSN* (not shown in figure) realizes up to 16% peak reduction for CP_1 , compared to only 12% with CP_2 . Note that today's datacenters are more in the CP_3 category, and these results reiterate the importance of eBuff/hybrid techniques for today's datacenters.

	Break-even point (in years)	Peak-width duration					
		30 minutes	1 hour	2 hours	3 hours	4 hours	8 hours
Lead-acid (3-5 years)	100 \$/KWh	0.3	0.7	1.4	2.1	2.8	5.6
	200 \$/KWh	0.7	1.4	2.8	4.2	5.6	11.1
	300 \$/KWh	1.0	2.1	4.2	6.3	8.3	16.7
VRB flow battery (12-15 years)	400 \$/KWh	1.4	2.8	5.6	8.3	11.1	22.2
	500 \$/KWh	1.7	3.5	6.9	10.4	13.9	27.8
	1000 \$/KWh	3.5	6.9	13.9	20.8	27.8	55.6

Figure 10: Cost-benefit analysis of investing on additional capacities for lead-acid and VRB batteries. Break-even points (in years) indicate the time it takes to recover the battery investment cost and start to make profit.

Investing In Additional Battery Capacity. We consider the cost-benefit of investing in additional battery capacities for both lead-acid batteries and newer energy-storage technologies such as *vanadium redox batteries* (VRBs). It is important to understand that these additional costs apply only to the battery packs and not the remaining circuitry of the UPS unit. We identify the procurement costs of these batteries for different energy storage capacities and find them to be 100-300\$/KWh for sealed lead-acid batteries and 400-500\$/KWh for VRB batteries [6]. While more expensive, VRBs have a longer inherent lifetime (12-15 years) compared to lead-acid batteries (3-5 years). We calculate break-even points - the time it takes to recover the battery

procurement cost and start to make profit - for workloads with different peak durations (ranging from 30 minutes to 8 hours) that are shaved using eBuff. In this analysis, we ignore lifetime concerns, and only include Op-ex gains - any Cap-ex gains would further hasten the occurrence of breaking even. We assume \$12/kW for peak tariff and present break-even points (in years) in Figure 10. The regions enclosed in solid and striped lines indicate scenarios where the break-even point occurs before the batteries have to be replaced due to their inherent lifetimes being reached. Additional battery capacity investments appears to stop offering benefits for workloads with eight or more hours of peak.

Summary of Key Insights.

- (i) Battery is preferable for tall and narrow peaks; throttling is preferable for wider and lower peaks.
- (ii) Cost benefits due to peak reduction significantly outweigh the cost incurred due to battery energy wastage.
- (iii) Hybrid technique combining battery and throttling do much better than using the knobs in isolation.
- (iv) Hybrid technique is able to dynamically adapt to different workload demands and SLAs.
- (v) Hybrid technique stands to gain substantially with any UPS over-provisioning (either in peak power or battery capacity) that may exist in today's datacenters. Even with aggressive provisioning going forward, hybrid techniques can provide better savings than pure throttling.
- (vi) Additional investments on UPS battery may prove profitable for Op-ex cost reduction for workloads with peak durations that are less than 4 hours.

7. RELATED WORK

Much of the early work on datacenter power focused on reducing the energy consumption by moving workloads off under-utilized servers and shutting them down [7–9, 14, 24, 38]. Numerous studies have also proposed techniques to trade-off performance with energy reduction [3, 10, 25, 33, 35, 46]. The impact of peak power on provisioning has been looked at from both provisioning for IT equipment - maximizing the utilization of the datacenter power infrastructure [12, 13, 19, 29, 36] as well as the cooling cost viewpoint [2, 42]. There have been a number of control techniques to cap power using some kind of throttling [16, 22, 40, 41, 45] and workload shaping [17, 34] temporally (moving the load over time to less critical points) and/or spatially (moving workloads across the datacenter). As mentioned earlier, our work proposes a complementary technique which can be used in conjunction with these earlier proposals. To our knowledge, storing energy for reduction of peak power needs of IT equipment is entirely novel. Closest to our domain is theoretical work on peak power reduction in residential buildings using batteries [4]. However, not being in the datacenter context, they ignore issues related to battery charge/discharges and lifetime, availability and provisioning concerns, which are very critical in determining the feasibility of stored energy for datacenter power management. An analogous idea which is used in buildings (and some datacenters) for cooling is thermal energy storage - such as brine chillers which generate ice during less expensive periods, for lowering demand during peak periods. Battery management has been primarily examined from the mobile/embedded context [15, 47] and in cyber-physical systems [48] where the drain rate is adjusted to elongate the time of operation.

8. CONCLUSION AND FUTURE WORK

The intention of this paper is to introduce the idea of energy buffers, and demonstrate its feasibility to reduce datacenters power costs. As opposed to previous techniques which attempt to shape the power consumption (either by workload shaping or throttling) to meet power availability, we have proposed a novel way by which we can do the reverse - storing the energy when available and using it to shape the power capacity so as to meet the consumption needs. We have shown how we can design mechanisms to exploit this idea given constraints of battery lifetimes, availability and electricity costs, and combine it with throttling based techniques in interesting ways. We have shown that hybrid approaches combining batteries and throttling can realize much higher performance for the desired cost savings across a spectrum of workloads and platform configurations using an experimental prototype. There are several directions for further research:

(i) Even though we have shown eBuff serving its purpose in existing current UPS configurations, we plan to investigate in detail, the cost-benefit trade-offs associated with (a) procuring larger battery capacities, (b) deploying vanadium-redox batteries with hardened lifetime, and, (c) using server-level UPS batteries (like in Google servers [18]) for Op-ex and Cap-ex reduction.

(ii) We can extend our hybrid (battery + throttling) techniques with workload shaping and scheduling for better dynamic adaptation. Further, we intend to investigate a mix of workloads having different SLAs co-existing in the data-center, with possibly different knobs being used for each.

(iii) Once we introduce the notion of energy storage, it opens the door to interesting topics for further study on how to manage this energy - what is the quantum (size) for management? can we allocate this to servers/applications? can we overbook this resource? what are the abstractions/system calls that the operating system should export? The storage feature of this resource gives a different perspective to these questions compared to traditional power management.

(iv) While the discussions and results in this paper are given specifically for battery-based UPS, there are ideas to extend from here to other UPS technologies, e.g. flywheels. When the energy storage capacity is not sufficient for more than a few seconds of operation, we note that captive diesel generators (DG) could step in to handle the peak loads. One should, of course, include the unit cost of such generation (which is typically around 1.5-2.0 times the cost of procurement from the utility) in developing such a mechanism. In fact, one could now envision a "multi-level energy buffer" management strategy spanning batteries and DGs.

(v) With the increasing push for renewable energy, such as solar/wind, the vagaries of availability makes energy storage even more critical. Hence, even if UPS units evolve into storage units that can sustain the load for only a few seconds, this work has a lot of applicability towards managing (possibly captive) renewable energy storage for datacenters.

Regardless of the extent of cost savings offered by eBuff, it is a knob that comes for relatively free - so why not use it?

Acknowledgements

This work was supported, in part, by the NSF grants 0811670, 0720456, 0615097, CAREER award 0953541, and a research award from HP

9. REFERENCES

- [1] 1 KW APC UPS - SURTA1500RML2U Runtime Charts. <http://www.apc.com/products/>.
- [2] F. Ahmad and T. N. Vijaykumar. Joint optimization of idle and cooling power in data centers while maintaining response time. In *Proceedings of the Architectural support for programming languages and operating systems (ASPLOS)*, 2010.
- [3] M. Annavaram, E. Grochowski, and J. Shen. Mitigating Amdahl's Law through EPI Throttling. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, 2005.
- [4] A. Bar-Noy, M. P. Johnson, and O. Liu. Peak Shaving Through Resource Buffering. In *Workshop On Approximation and Online Algorithms (WAOA)*, 2008.
- [5] L. A. Barroso and U. Holzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Morgan and Claypool Publishers, 2009.
- [6] Battery costs. http://photovoltaics.sandia.gov/Pubs_2010/PV%20Website%20Publications%20Folder_09/Hanley_PVSC09%5B1%5D.pdf.
- [7] P. Bohrer, D. Cohn, E. Elnozahy, T. Keller, M. Kistler, C. Lefurgy, R. Rajamony, F. Rawson, and E. V. Hensbergen. Energy Conservation for Servers. In *Workshop on Power Management for Real-Time and Embedded Systems*, 2001.
- [8] J. Chase, D. Anderson, P. Thakur, and A. Vahdat. Managing Energy and Server Resources in Hosting Centers. In *Proceedings of the Symposium on Operating Systems Principles (SOSP)*, 2001.
- [9] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *Proceedings of Networked Systems Design and Implementation (NSDI)*, 2008.
- [10] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam. Managing Server Energy and Operational Costs in Hosting Centers. In *Proceedings of the Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2005.
- [11] Duke utility bill tariff. <http://www.duke-energy.com/pdfs/scscheduleopt.pdf>.
- [12] X. Fan, W.-D. Weber, and L. A. Barroso. Power Provisioning for a Warehouse-Sized Computer. In *Proceedings of the Annual International Symposium on Computer Architecture (ISCA)*, 2007.
- [13] W. Felter, K. Rajamani, C. Rusu, and T. Keller. A Performance-Conserving Approach for Reducing Peak Power Consumption in Server Systems. In *Proceedings of the International Conference on Supercomputing (ICS)*, 2005.
- [14] M. E. Femal and V. W. Freeh. Safe overprovisioning: Using power limits to increase aggregate throughput. In *Workshop on Power-Aware Computer Systems (PACS)*, 2004.
- [15] J. Flinn and M. Satyanarayanan. Managing battery lifetime with energy-aware adaptation. *Transaction on Computer Systems (TOCS)*, 2004.
- [16] A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy. Optimal power allocation in server farms.

- In *Proceedings of the Conference on Measurement and Modeling of Computer Systems (SIGMETRICS)*, 2009.
- [17] L. Ganesh, J. Liu, S. Nath, G. Reeves, and F. Zhao. Unleash stranded power in data centers with rackpacker. In *Workshop on Energy-Efficient Design (WEED)*, 2009.
 - [18] Google Server-level UPS for improved efficiency. http://news.cnet.com/8301-1001_3-10209580-92.html.
 - [19] S. Govindan, J. Choi, B. Urgaonkar, A. Sivasubramaniam, and A. Baldini. Statistical profiling-based techniques for effective power provisioning in data centers. In *Proceedings of european conference on Computer systems (EUROSYS)*, 2009.
 - [20] S. Govindan, D. Wang, L. Y. Chen, A. Sivasubramaniam, and B. Urgaonkar. Modeling and Analysis of Availability of Datacenter Power Infrastructure. Technical Report CSE-10-006, The Pennsylvania State University, 2010.
 - [21] J. Hamilton. Internet-scale Service Infrastructure Efficiency, ISCA Keynote 2009.
 - [22] T. Heath, A. P. Centeno, P. George, L. Ramos, Y. Jaluria, and R. Bianchini. Mercury and Freon: Temperature Emulation and Management in Server Systems. In *Proceedings of Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2006.
 - [23] J. Hellerstein. Google Cluster Data, 2010. <http://googleresearch.blogspot.com/2010/01/google-cluster-data.html>.
 - [24] T. Horvath and K. Skadron. Multi-mode energy management for multi-tier server clusters. In *Proceedings of the Parallel Architectures and Compilation Techniques (PACT)*, 2008.
 - [25] C. Isci, A. Buyuktosunoglu, C.-Y. Cher, P. Bose, and M. Martonosi. An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget. In *Proceedings of the Symposium on Microarchitecture (MICRO)*, 2006.
 - [26] A. Kleywegt, V. Nori, M. Savelsbergh, and C. A. Tovey. Online resource minimization. In *Symposium on Discrete algorithms (SODA)*, 1999.
 - [27] K. Le, R. Bianchini, M. Martonosi, and T. Nguyen. Cost- and energy-aware load distribution across data centers. In *Workshop on Power-Aware Computing and Systems (HOTPOWER)*, 2009.
 - [28] Lead-acid batteries: Lifetime vs Depth of discharge. www.windsun.com/Batteries/Battery_FAQ.htm.
 - [29] C. Lefurgy, X. Wang, and M. Ware. Server-Level Power Control. In *International Conference on Autonomic Computing (ICAC)*, 2007.
 - [30] B. Li, G. Keung, S. Xie, F. Liu, Y. Sun, and H. Yin. An empirical study of flash crowd dynamics in a p2p-based live video streaming system. In *Global Telecommunications Conference (GLOBECOM)*, 2008.
 - [31] D. Linden and T. B. Reddy. *Handbook of Batteries*. McGraw Hill Handbooks, 2002.
 - [32] S. McCluer. APC White paper 30: Battery Technology for Data Centers and Network Rooms, 2005.
 - [33] D. Meisner, B. T. Gold, and T. F. Wenisch. Powernap: eliminating server idle power. In *Proceedings of the Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2009.
 - [34] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making Scheduling Cool: Temperature-Aware Workload Placement in Data Centers. In *Proceedings of the Usenix Annual Technical Conference (USENIX)*, 2005.
 - [35] R. Nathuji and K. Schwan. Virtualpower: Coordinated power management in virtualized enterprise systems. In *Proceedings of the Symposium on Operating Systems Principles (SOSP)*, 2007.
 - [36] S. Pelley, D. Meisner, P. Zandevakili, T. F. Wenisch, and J. Underwood. Power Routing: Dynamic Power Provisioning in the Data Center. In *Proceedings of the Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2010.
 - [37] PGE utility bill tariff. www.pge.com/tariffs/CommercialCurrent.xls.
 - [38] E. Pinheiro, R. Bianchini, E. Carrera, and T. Heath. Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems. In *Workshop on Compilers and Operating Systems for Low Power (COLP)*, 2001.
 - [39] Puget Energy: Time-of-use electricity billing. http://energypriorities.com/entries/2006/02/pse_tou_amr_case.php.
 - [40] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu. No Power Struggles: Coordinated multi-level power management for the data center. In *Proceedings of Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2008.
 - [41] L. Ramos and R. Bianchini. C-Oracle: Predictive thermal management for data centers. In *High Performance Computer Architecture (HPCA)*, 2008.
 - [42] P. Ranganathan, P. Leech, D. Irwin, and J. Chase. Ensemble-level Power Management for Dense Blade Servers. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, 2006.
 - [43] SPEC JBB2005: Java Business Benchmark. <http://www.spec.org/jbb2005/>.
 - [44] A. Vasani, A. Sivasubramaniam, V. Shimpi, T. Sivabalan, and R. Subbiah. Worth their watts? - an empirical study of datacenter servers. In *Proceedings of High Performance Computer Architecture (HPCA)*, 2010.
 - [45] X. Wang and M. Chen. Cluster-level feedback power control for performance optimization. In *High Performance Computer Architecture (HPCA)*, 2008.
 - [46] A. Weisel and F. Bellosa. Process cruise control: Event-driven clock scaling for dynamic power management. In *Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, 2002.
 - [47] H. Zeng, X. Fan, C. Ellis, A. Lebeck, and A. Vahdat. ECOSystem: Managing Energy as a First Class Operating System Resource. In *Proceedings of the Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2002.
 - [48] F. Zhang, Z. Shi, and W. Wolf. A dynamic battery model for co-design in cyber-physical systems. In *Proceedings of the Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2009.